

REMARKS

The Office Action dated August 12, 2004 has been reviewed carefully and the application has been amended in a sincere effort to place it in condition for allowance. Continued prosecution is requested in the accompanying Request for Continued Examination.

Claim Rejections - 35 U.S.C. §112

Claims 8 and 11 were rejected under 35 U.S.C. §112 first paragraph. Claims 8 and 11 have been amended herein to clarify that the pseudo header is inserted as octets of data after the protocol header, before the protocol data in the data field, as stated in the Specification at p. 19, lines 5-6, "consistent with the elements of the present invention, the data field may be constrained to carry a pseudo header field." Applicant respectfully submits that the claims are now in condition for allowance and reconsideration of the rejection in this regard is requested.

Claims Rejections - 35 U.S.C. §103

Claims 1-17 and 21-30 were rejected under 35 U.S.C. §103 as being obvious over United States Patent No. 6,590,903 to Hofers et al. ("Hofers").

Briefly, the present invention enables two devices that employ dislike protocols to effectively communicate with each other. By way of example, and not as a limitation on the scope of the invention, the first host, in one example, may have a UDP engine, while the second host, in one example, has a TCP engine. By inserting an additional header in

packets constructed and generated in accordance with the invention, we have effectively persuaded the second host with its TCP engine to accept a packet formatted in primary accordance with the UDP protocol by the first host.

Thus, in accordance with the present invention, only one end of the communication has to be modified. The other end of the communication remains unmodified and in essence is unaware of the invention being used at the first end, and thus needs to implement only one protocol suite. This results in resource savings at the second host.

In contrast, as observed by the Examiner, Hofers teaches a method of protocol conversion. In protocol conversion, as in Hofers, one protocol (for example an asynchronous data traffic produced by a microprocessor) [Hofers Col. 2, line 55] is converted into a synchronous protocol (for example an I2S bus used to transmit data frames in time multiplex). [Hofers Col. 1, line 10].

Further in Hofers, two instances of the communication apparatus or circuitry [Col2, line 50-60] sit between the two microprocessors at either end. Each such apparatus or conversion circuit needs to be programmed to add a header at the transmission end, to co-ordinate and serve as markers of the time multiplexed communication [Col1, line 10]. The other end circuit strips this header, its purpose having been served. Thus, (a) both ends are performing their activities in true accordance with a single protocol their owe allegiance to; and (b) the header is only used to encapsulate the data in the link in between.

In Applicant's invention, there is no protocol conversion. Instead, in one exemplary embodiment of the invention, for example, as described in the instant Specification

at page 19, Section "Exemplary processing," data is transmitted by one host in primary in accordance with one protocol and received by the peer host in primary accordance with the procedures of a second protocol.

In the exemplary embodiment, the first host could be the network peer 120, and the second host could be a communication device containing an embodiment of the present invention, for example. In such an exemplary embodiment, the first host could be transmitting a packet in primary accordance with UDP while the second host is performing reception processing of the same sent packet, now received, but using its implementation of a second protocol, for example TCP. There is no translation or conversion, the second device simply digests the packet as a TCP packet.

More specifically, the techniques and system of the present invention involve inserting an additional header after the protocol header, at the first host. Thus it is possible for the second host to process the packet formatted by the first host in primary (in accordance with a first protocol, for example UDP), but with an additional inserted header, which allows the second host to consume the packet as if it were in accordance with the second protocol.

This additional header, inserted after the header mandated by the first protocol, (e.g., UDP in this example), allows the second host to be able to use its existing implementation of a second protocol, (e.g., TCP in this example), to process the packet, without protocol conversion. If this additional header were absent, the second host will have no choice but to discard the packet since the packet does not conform to its primary expectation of its protocol in this instance, namely TCP.

The first host, while formatting its packet in primary accordance with the first protocol, namely UDP, has been taught by the programming in accordance with the teachings of this invention, to add an additional header, after its protocol header in primary accordance with its obligations to its protocol of allegiance, namely UDP in this example. This additional header allows the second host to process the very same packet in primary accordance with its own protocol of allegiance, namely TCP in this example.

In accordance with the invention, this additional header, inserted after the required primary protocol header, is called a "pseudo-header". We call this additional header a "pseudo-header" because, in so far as the rules of the first protocol, namely UDP, which is the protocol of allegiance at the first host, this additional header is treated as part of the data transmitted and not treated as header. However, at the receiving end, it is treated as a header. Hence, we have termed this portion of the packet, a pseudo-header.

It is noted that in yet another embodiment, the roles of 110 and 120 may be reversed. Still further embodiments can employ other protocols being substituted in place of UDP and TCP.

The type of solution for devices operating under different protocols provided by the present invention is a solution that was not known prior to this invention. In conventional designs, both hosts 110 and 120 were required to implement the same protocol as the other end in order to be able to communicate with each other. Packets sent using UDP by one host would simply be rejected by the other host if it only implemented TCP, without a Hofer's type of protocol conversion at both ends.

Using the techniques and system of the present invention, by means of inserting an additional header after the protocol header, at the first host, we make it possible for the second host to process the packet formatted by the first host in primary in accordance with a first protocol, for example UDP, but with an additional inserted header, which allows the second host to consume the packet as if it were in accordance with the second protocol, without additional conversion programming or other software processes.

As noted, the additional header, inserted after the header mandated by the first protocol, (e.g., UDP in this example), allows the second host to be able to use its existing implementation of a second protocol, (e.g., TCP in this example), to process the packet, without protocol conversion.

The present invention primarily teaches enablement of parsing packets formatted in accordance with a first protocol at a first host using a protocol processing means or engine at a second host with primary design to accept packets using a second protocol. The ability to respond to such a packet is also a feature of the present invention; in the response, the roles of first host and second host, and first protocol and second protocol, get reversed.

As observed by the examiner, the act of replying in and of itself is a known technique, however, the act of applying the techniques of the present invention as described is not known or anticipated. In other words, Hofers and the other cited references herein also do not describe creating and inserting a pseudo-header, in the reverse direction to allow (in the example) the second host with a TCP engine to communicate back to the first host with, in this example, a UDP engine.

Accordingly, the Hofers reference does not disclose, teach or anticipate Applicants invention as recited in the claims.

With respect to claims 2-17 and 22-20, these claims dependent directly or indirectly from independent claims 1 and 21. Independent Claim 1 has been amended, and Claim 21 has been distinguished herein over the cited references. Applicant respectfully submits that the independent claims and the claims dependent there from are allowable over the cited art.

Claims 18 – 20 were also rejected as obvious over the Hofers reference. The Examiner indicated that while Hofers does not identify a “prefix” the teachings render the concept obvious. However, Applicant’s usage of the word “prefix” relates to the result of analyzing the commonality between two different protocols, one of which is implemented at one end of the line, at the first host, and the other of which is implemented at the other end of the line, namely the second host. In one example, the first protocol is UDP while the second protocol is TCP.

In accordance with the present invention, the "prefix" refers to the portion of the header of UDP and TCP, in this example, that are common to both protocols. Applicant’s draw attention to the description depicted in the Specification at page 19, Section "Exemplary processing," and in the accompanying Figure 7, with protocol elements marked by name (being exactly the same of the first protocol, namely UDP) and the protocol elements in the pseudo-header, marked by labels 710 through 760. The description on page 19 through 20 teaches how the prefix is formatted as per prior-art, while the ad-

ditional header (i.e. the suffix), namely the pseudo-header with elements labeled 710 through 760, is formatted in accordance with the additional constraints of our invention.

On the other hand, Hofers does not teach the comparison of one protocol against the other to formulate a prefix and a pseudo header. The teachings of Hofers convert one protocol into the other; it converts an asynchronous protocol into synchronous so that a like receiver, namely a synchronous receiver, at the second end will receive it.

In accordance with the present invention, two dislike protocols to effectively communicate with each other. The first host, in one example, has a UDP engine, while the second host, in one example, has a TCP engine. By inserting an additional header, we have persuaded the second host with its TCP engine to accept a packet formatted in primary accordance with the UDP protocol by the first host.

Stated another way, Hofers has two host which are symmetrical and complementary to each other. The present invention contemplates two hosts with dislike protocol implementations, and – absent the invention – due to their dissymmetry, they would be incapable of accepting each other's packets and hence incapable of communicating with each other.

As discussed, the header of the packet is examined by the recipient as if it were written in the second protocol. This is not taught disclosed or rendered obvious by Hofers for the reasons outlined above. Hofers teaches an apparatus or conversion circuit

that is inserted at both ends of communication medium. At the first end, the conversion circuit places the data from the asynchronous communication, whilst adding a header, into the synchronous communication channel. However, the other end in the case of Hofers, is also provided with a companion conversion circuit of the same invention, that does the reverse, namely it recognizes the header, removes it and places the asynchronous data back in its native form. This is encapsulation that converts one protocol into another, and at the other end, the corresponding de-capsulation which converts the data inside the second protocol back to the first.

In the present invention, only one end of the communication has been modified. The other end of the communication remains unmodified and in essence is unaware of the invention being used at the first end, and thus needs to implement only one protocol suite. This results in resource saving at the second host.

In view of these distinctions, it is believed that claims 18 -20 are allowable over the cited reference. Claims 19 and 20 depend directly or indirectly upon Claim 18.

Claims 1, 8, 11 and 21 were rejected under 35 U.S.C. 103(a) as being obvious over printed publication "Open System Interconnect (OSI) Model" ("Brown") as cited by the Examiner. Applicant respectfully submits that in view of the amendments to the claims herein, and based upon the above arguments, the Brown reference does not render the claimed invention obvious. Specifically, Hofers teaches encapsulation, and Brown teaches encapsulation. However, encapsulation is symmetrical, the first end of the communication does the encapsulation, and the other end undoes it. This does not suggest or render obvious Applicants invention in which a packet is sent in one format, and auto-

matically received in another at the second end, with only one end requiring modification.

With respect to the Examiners Response to Applicant's arguments, Applicant points out that the 112 rejections have been addressed herein. With respect to the arguments regarding Hofers, the present invention is not performing a protocol conversion, but is using code in one protocol that is understood by the recipient engine to be in accordance with a second protocol. Thus, Hofers' suggestion of a stop bit and word count does not disclose, teach or suggest Applicant's invention.

SUMMARY

All of the objections and rejections have been addressed herein. Accordingly, Applicant respectfully submits that the Application is in condition for allowance.

Please do not hesitate to contact the undersigned in order to advance the prosecution of this application in any respect.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,



Rita M. Rooney
Reg. No. 30,585
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500